

# Autour du traitement numérique du signal

MP\* 2023-2024 LVH

## Plan : I - Filtrage numérique

- Attendus du programme
- Passage de la fonction de transfert à une relation de récurrence



- Mettre en œuvre un **convertisseur analogique/numérique**, aussi appelé **CAN** ;

- Mettre en œuvre un **convertisseur analogique/numérique**, aussi appelé **CAN** ;
- on utilisera pour cela la carte d'acquisition SYSAM-SP5 (connue également sous le petit nom de Marguerite...);

- Mettre en œuvre un **convertisseur analogique/numérique**, aussi appelé **CAN** ;
- on utilisera pour cela la carte d'acquisition SYSAM-SP5 (connue également sous le petit nom de Marguerite...);
- et **un traitement numérique** afin de réaliser un filtre passe-bas ;

- Mettre en œuvre un **convertisseur analogique/numérique**, aussi appelé **CAN** ;
- on utilisera pour cela la carte d'acquisition SYSAM-SP5 (connue également sous le petit nom de Marguerite...);
- et **un traitement numérique** afin de réaliser un filtre passe-bas ;
- on va le programmer en python ;

- Mettre en œuvre un **convertisseur analogique/numérique**, aussi appelé **CAN** ;
- on utilisera pour cela la carte d'acquisition SYSAM-SP5 (connue également sous le petit nom de Marguerite...);
- et un **traitement numérique** afin de réaliser un filtre passe-bas ;
- on va le programmer en python ;
- utiliser un **convertisseur numérique/analogique CNA** pour restituer un signal analogique.

- Mettre en œuvre un **convertisseur analogique/numérique**, aussi appelé **CAN** ;
- on utilisera pour cela la carte d'acquisition SYSAM-SP5 (connue également sous le petit nom de Marguerite...);
- et un **traitement numérique** afin de réaliser un filtre passe-bas ;
- on va le programmer en python ;
- utiliser un **convertisseur numérique/analogique CNA** pour restituer un signal analogique.
- on utilise également la carte d'acquisition SYSAM-SP5

## Plan : I - Filtrage numérique

- Attendus du programme
- Passage de la fonction de transfert à une relation de récurrence

- Fonction de transfert passe-bas premier ordre, gain maximal  $G_0 = 1$ , pulsation de coupure  $\omega_c = 2\pi f_c$  :

- Fonction de transfert passe-bas premier ordre, gain maximal  $G_0 = 1$ , pulsation de coupure  $\omega_c = 2\pi f_c$  :

- 

$$\underline{H}(j\omega) = \frac{1}{1 + j\frac{\omega}{\omega_c}}$$

- Fonction de transfert passe-bas premier ordre, gain maximal  $G_0 = 1$ , pulsation de coupure  $\omega_c = 2\pi f_c$  :

- $$H(j\omega) = \frac{1}{1 + j\frac{\omega}{\omega_c}}$$

- Domaine temporel :

- Fonction de transfert passe-bas premier ordre, gain maximal  $G_0 = 1$ , pulsation de coupure  $\omega_c = 2\pi f_c$  :

- $$H(j\omega) = \frac{1}{1 + j\frac{\omega}{\omega_c}}$$

- Domaine temporel :

- $$\frac{1}{\omega_c} \frac{ds}{dt} + s(t) = e(t),$$

- Fonction de transfert passe-bas premier ordre, gain maximal  $G_0 = 1$ , pulsation de coupure  $\omega_c = 2\pi f_c$  :

- 

$$\underline{H}(j\omega) = \frac{1}{1 + j\frac{\omega}{\omega_c}}$$

- Domaine temporel :

- 

$$\frac{1}{\omega_c} \frac{ds}{dt} + s(t) = e(t),$$

- en notant  $e(t)$  le signal d'entrée et  $s(t)$  le signal de sortie.

- Fonction de transfert passe-bas premier ordre, gain maximal  $G_0 = 1$ , pulsation de coupure  $\omega_c = 2\pi f_c$  :

- 

$$H(j\omega) = \frac{1}{1 + j\frac{\omega}{\omega_c}}$$

- Domaine temporel :

- 

$$\frac{1}{\omega_c} \frac{ds}{dt} + s(t) = e(t),$$

- en notant  $e(t)$  le signal d'entrée et  $s(t)$  le signal de sortie.
- Étape importante : **discrétisation** de l'équation différentielle.

- Seuls instants disponibles : les  $t_k = kT_e$

- Seuls instants disponibles : les  $t_k = kT_e$
- $e(t)$  remplacé par  $e_k = e(kT_e)$

- Seuls instants disponibles : les  $t_k = kT_e$
- $e(t)$  remplacé par  $e_k = e(kT_e)$
- $s(t)$  par  $s_k = s(kT_e)$

- Seuls instants disponibles : les  $t_k = kT_e$
- $e(t)$  remplacé par  $e_k = e(kT_e)$
- $s(t)$  par  $s_k = s(kT_e)$

- 

$$\frac{ds}{dt} = \frac{s(t + dt) - s(t)}{dt} = ?$$

- Seuls instants disponibles : les  $t_k = kT_e$
- $e(t)$  remplacé par  $e_k = e(kT_e)$
- $s(t)$  par  $s_k = s(kT_e)$

- 

$$\frac{ds}{dt} = \frac{s(t + dt) - s(t)}{dt} = ?$$

- $dt$  remplacé par  $T_e$  (c'est la plus petite durée que l'on manipule)

- Seuls instants disponibles : les  $t_k = kT_e$
- $e(t)$  remplacé par  $e_k = e(kT_e)$
- $s(t)$  par  $s_k = s(kT_e)$

- 

$$\frac{ds}{dt} = \frac{s(t + dt) - s(t)}{dt} = ?$$

- $dt$  remplacé par  $T_e$  (c'est la plus petite durée que l'on manipule)
- d'où

$$\frac{ds}{dt}(t_k) \simeq \frac{s(t_k + T_e) - s(t_k)}{T_e} = \frac{s_{k+1} - s_k}{T_e}.$$

- Pour finir : l'équation différentielle discrétisée est

- Pour finir : l'équation différentielle discrétisée est

- 

$$\frac{s_{k+1} - s_k}{\omega_c T_e} + s_k = e_k$$

- Pour finir : l'équation différentielle discrétisée est



$$\frac{s_{k+1} - s_k}{\omega_c T_e} + s_k = e_k$$

- Soit

$$s_{k+1} = s_k + \omega_c T_e (e_k - s_k).$$

- Pour finir : l'équation différentielle discrétisée est



$$\frac{s_{k+1} - s_k}{\omega_c T_e} + s_k = e_k$$

- Soit

$$s_{k+1} = s_k + \omega_c T_e (e_k - s_k).$$

- Relation de récurrence permettant, puisque l'on connaît les  $e_k$ , de calculer de proche en proche tous les  $s_k$ , en prenant une condition initiale, par exemple :  $s_0 = 0$ .

- Pour finir : l'équation différentielle discrétisée est



$$\frac{s_{k+1} - s_k}{\omega_c T_e} + s_k = e_k$$

- Soit

$$s_{k+1} = s_k + \omega_c T_e (e_k - s_k).$$

- Relation de récurrence permettant, puisque l'on connaît les  $e_k$ , de calculer de proche en proche tous les  $s_k$ , en prenant une condition initiale, par exemple :  $s_0 = 0$ .
- Programmation élémentaire...

## Plan : II - Manipulation

- Remarques préliminaires...
- Description du fichier FiltrageNumerique.py
- Ce que vous devez faire. Echauffement...
- Ce que vous devez faire. Course de fond...
- Ce que vous devez faire. Décrassage s'il reste du temps...

- Idée : on va commander l'acquisition et la génération d'un signal par la carte SYSAM-SP5 directement par un programme python

- Idée : on va commander l'acquisition et la génération d'un signal par la carte SYSAM-SP5 directement par un programme python
- Cependant pour ce faire il faut utiliser des versions compatibles du module pycanum et de python, à savoir pour nous python 3.8.10

- Idée : on va commander l'acquisition et la génération d'un signal par la carte SYSAM-SP5 directement par un programme python
- Cependant pour ce faire il faut utiliser des versions compatibles du module pycanum et de python, à savoir pour nous python 3.8.10
- Pour lancer pyzo avec le bon python, aller sur le disque L :

- Idée : on va commander l'acquisition et la génération d'un signal par la carte SYSAM-SP5 directement par un programme python
- Cependant pour ce faire il faut utiliser des versions compatibles du module pycanum et de python, à savoir pour nous python 3.8.10
- Pour lancer pyzo avec le bon python, aller sur le disque L :
- Naviguer jusqu'au répertoire

L:\Maths\_SG\TPTraitementSignal\PyzoPortable

- Idée : on va commander l'acquisition et la génération d'un signal par la carte SYSAM-SP5 directement par un programme python
- Cependant pour ce faire il faut utiliser des versions compatibles du module pycanum et de python, à savoir pour nous python 3.8.10
- Pour lancer pyzo avec le bon python, aller sur le disque L :
- Naviguer jusqu'au répertoire

`L:\Maths_SG\TPTraitementSignal\PyzoPortable`

- Double-cliquez sur l'icône pyzo. Il est probable qu'il vous sera demandé de configurer le shell... Cliquez sur le lien correspondant et dans la fenêtre exe taper (ou copier-coller à partir du contenu du fichier texte sur le site) :

`L:\Maths_SG\TraitementSignalMP\App\python.exe`

## Plan : II - Manipulation

- Remarques préliminaires...
- Description du fichier FiltrageNumerique.py
- Ce que vous devez faire. Echauffement...
- Ce que vous devez faire. Course de fond...
- Ce que vous devez faire. Décrassage s'il reste du temps...

- Fichier FiltrageNumerique.py sur le site du lycée.

- Fichier FiltrageNumerique.py sur le site du lycée.
- Il se compose de plusieurs cellules :

- Fichier FiltrageNumerique.py sur le site du lycée.
- Il se compose de plusieurs cellules :
- La première concerne tous les imports nécessaires. Il faudra bien sûr l'exécuter avant tous les autres

- Fichier FiltrageNumerique.py sur le site du lycée.
- Il se compose de plusieurs cellules :
- La première concerne tous les imports nécessaires. Il faudra bien sûr l'exécuter avant tous les autres
- La deuxième est une cellule à exécuter pour faire l'acquisition d'un signal sur l'entrée EA0.

- Fichier FiltrageNumerique.py sur le site du lycée.
- Il se compose de plusieurs cellules :
- La première concerne tous les imports nécessaires. Il faudra bien sûr l'exécuter avant tous les autres
- La deuxième est une cellule à exécuter pour faire l'acquisition d'un signal sur l'entrée EA0.
- La troisième contient une fonction passebas à compléter selon la docstring

- Fichier FiltrageNumerique.py sur le site du lycée.
- Il se compose de plusieurs cellules :
- La première concerne tous les imports nécessaires. Il faudra bien sûr l'exécuter avant tous les autres
- La deuxième est une cellule à exécuter pour faire l'acquisition d'un signal sur l'entrée EA0.
- La troisième contient une fonction passebas à compléter selon la docstring
- La quatrième est la cellule à exécuter pour générer un signal sur la sortie SA1.

## Plan : II - Manipulation

- Remarques préliminaires...
- Description du fichier FiltrageNumerique.py
- **Ce que vous devez faire. Echauffement...**
- Ce que vous devez faire. Course de fond...
- Ce que vous devez faire. Décrassage s'il reste du temps...

- Récupérer le fichier FiltrageNumerique.py sur le site du lycée.

- Récupérer le fichier FiltrageNumerique.py sur le site du lycée.
- L'ouvrir avec Pyzo en shell python 3.8.10

- Récupérer le fichier FiltrageNumerique.py sur le site du lycée.
- L'ouvrir avec Pyzo en shell python 3.8.10
- Écrire la fonction passebas conformément aux spécifications de la docstring

- Récupérer le fichier FiltrageNumerique.py sur le site du lycée.
- L'ouvrir avec Pyzo en shell python 3.8.10
- Écrire la fonction passebas conformément aux spécifications de la docstring
- Faire l'acquisition d'un signal sinusoïdal, d'amplitude de quelques volts, de fréquence 100 Hz (Réfléchir aux paramètres d'acquisition...)

- Récupérer le fichier FiltrageNumerique.py sur le site du lycée.
- L'ouvrir avec Pyzo en shell python 3.8.10
- Écrire la fonction passebas conformément aux spécifications de la docstring
- Faire l'acquisition d'un signal sinusoïdal, d'amplitude de quelques volts, de fréquence 100 Hz (Réfléchir aux paramètres d'acquisition...)
- Fabriquer les différents signaux correspondant au filtrage de ce signal par un passe-bas du premier ordre de fréquence de coupure 10 Hz, 100 Hz, 1000 Hz.

- Récupérer le fichier FiltrageNumerique.py sur le site du lycée.
- L'ouvrir avec Pyzo en shell python 3.8.10
- Écrire la fonction passebas conformément aux spécifications de la docstring
- Faire l'acquisition d'un signal sinusoïdal, d'amplitude de quelques volts, de fréquence 100 Hz (Réfléchir aux paramètres d'acquisition...)
- Fabriquer les différents signaux correspondant au filtrage de ce signal par un passe-bas du premier ordre de fréquence de coupure 10 Hz, 100 Hz, 1000 Hz.
- Générer successivement sur la sortie SA1 les signaux correspondants aux résultats des filtrages et observer les à l'oscilloscope

- Représenter sur un même graphe le signal original ainsi qu'un seul signal filtré (à faire successivement avec chaque signal filtré)

- Représenter sur un même graphe le signal original ainsi qu'un seul signal filtré (à faire successivement avec chaque signal filtré)
- Commenter abondamment les courbes. Représenter éventuellement uniquement le "début" des signaux...

- Représenter sur un même graphe le signal original ainsi qu'un seul signal filtré (à faire successivement avec chaque signal filtré)
- Commenter abondamment les courbes. Représenter éventuellement uniquement le "début" des signaux...
- Représenter sur un même graphe le signal original ainsi que tous les signaux filtrés.

## Plan : II - Manipulation

- Remarques préliminaires...
- Description du fichier FiltrageNumerique.py
- Ce que vous devez faire. Echauffement...
- **Ce que vous devez faire. Course de fond...**
- Ce que vous devez faire. Décrassage s'il reste du temps...

- Première manipulation

- Première manipulation
- Faire l'acquisition d'un signal sinusoïdal de fréquence 50 Hz, d'amplitude 4 V, à nommer **signal**

- Première manipulation
- Faire l'acquisition d'un signal sinusoïdal de fréquence 50 Hz, d'amplitude 4 V, à nommer **signal**
- Faire l'acquisition d'un signal rectangulaire de fréquence 1000 Hz, d'amplitude 100 mV, à nommer **bruit**

- Première manipulation
- Faire l'acquisition d'un signal sinusoïdal de fréquence 50 Hz, d'amplitude 4 V, à nommer **signal**
- Faire l'acquisition d'un signal rectangulaire de fréquence 1000 Hz, d'amplitude 100 mV, à nommer **bruit**
- Fabriquer le signal bruité, nommé **signalbruite**, à partir des deux signaux précédents

- Première manipulation
- Faire l'acquisition d'un signal sinusoïdal de fréquence 50 Hz, d'amplitude 4 V, à nommer **signal**
- Faire l'acquisition d'un signal rectangulaire de fréquence 1000 Hz, d'amplitude 100 mV, à nommer **bruit**
- Fabriquer le signal bruité, nommé **signalbruite**, à partir des deux signaux précédents
- Filtrer ce signal bruité pour retrouver le signal initial. Le résultat sera appelé **signalfiltre**

- Première manipulation
- Faire l'acquisition d'un signal sinusoïdal de fréquence 50 Hz, d'amplitude 4 V, à nommer **signal**
- Faire l'acquisition d'un signal rectangulaire de fréquence 1000 Hz, d'amplitude 100 mV, à nommer **bruit**
- Fabriquer le signal bruité, nommé **signalbruite**, à partir des deux signaux précédents
- Filtrer ce signal bruité pour retrouver le signal initial. Le résultat sera appelé **signalfiltre**
- Représenter les deux signaux **signalbruite** et **signalfiltre** sur la même fenêtre pour apprécier le résultat du filtrage.

- Deuxième manipulation

- Deuxième manipulation
- Écrire dans le fichier FiltrageNumerique.py une fonction **passerhaut** dans la section dédiée au filtrage, réalisant un filtrage numérique de type passe-haut du premier ordre de fréquence de coupure  $f_c$ .

- Deuxième manipulation
- Écrire dans le fichier FiltrageNumerique.py une fonction **passerhaut** dans la section dédiée au filtrage, réalisant un filtrage numérique de type passe-haut du premier ordre de fréquence de coupure  $f_c$ .
- tester votre fonction sur différents signaux et visualiser le résultat à l'oscilloscope ou par représentation graphique (matplotlib)

## Plan : II - Manipulation

- Remarques préliminaires...
- Description du fichier FiltrageNumerique.py
- Ce que vous devez faire. Echauffement...
- Ce que vous devez faire. Course de fond...
- Ce que vous devez faire. Décrassage s'il reste du temps...

- Au choix :

- Au choix :
- Créer un signal bruité avec un bruit aléatoire

- Au choix :
- Créer un signal bruité avec un bruit aléatoire
- Tester l'influence d'un filtrage de type passe bas ou passe-haut sur un signal rectangulaire

- Au choix :
- Créer un signal bruité avec un bruit aléatoire
- Tester l'influence d'un filtrage de type passe bas ou passe-haut sur un signal rectangulaire
- Essayer d'implémenter des filtres d'ordre 2, par exemple un passe-bande